NORQUEST COLLEGE

CMPT1011

Introduction to Computing

Unit 7: Strings

Learning Outcome

- Explain that string is a sequence
- Apply slicing to access a segment of a string
- Explain that strings are immutable

Learning Outcome

- Demonstrate string operators in print function and the 'in' operator

- Demonstrate string methods e.g. upper, lower, 'isalnum', find, etc.

Strings

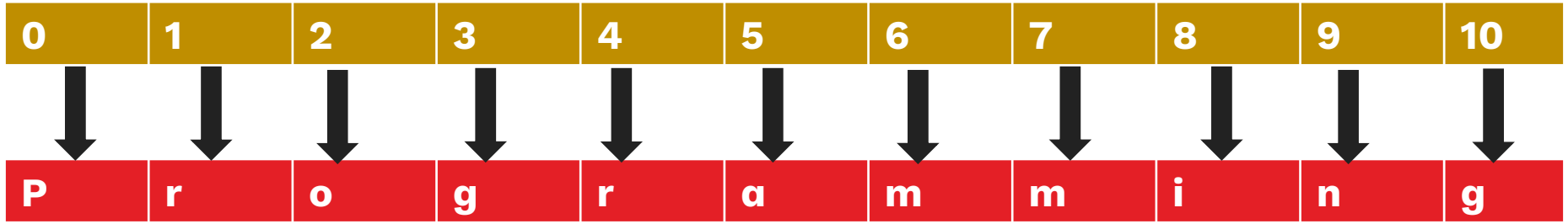
Strings

A string is a sequence of characters enclosed in either single or double quotation marks. E.g.,

1. “I am a string”
2. ‘Hello’
3. “Welcome to Python Prog”

Strings

- Strings can also be referred to as collection of characters.
- These characters can be accessed individually.



Accessing Strings

- You can access the characters by using [] brackets.

e.g.

school = ‘norquest’

letter = school[2]

- The number in brackets is referred to as index.

Immutable

Immutable

- Python strings are immutable, which means they cannot be changed, once they are created.

- Since strings can't be changed, we construct ***new*** strings as we go to represent computed values.

Example

school = ‘norquest’
school[2] = ‘w’

Output:
TypeError: 'str' object does not support item assignment

Length function: len()

- The len() function is used to find the length of characters.

```
school = ‘norquest’
len(school)
```

Output:

8

Slicing

Slice() function

- A segment of a string is called a slice.
- We can slice a string and save it in another variable or print it.
- Selecting a slice is similar to selecting a character.

Syntax

String[n:m] e.g name[0:7]

returns the part of string from the “n-eth” character to the “m-eth” character, **including the first but excluding the last.**

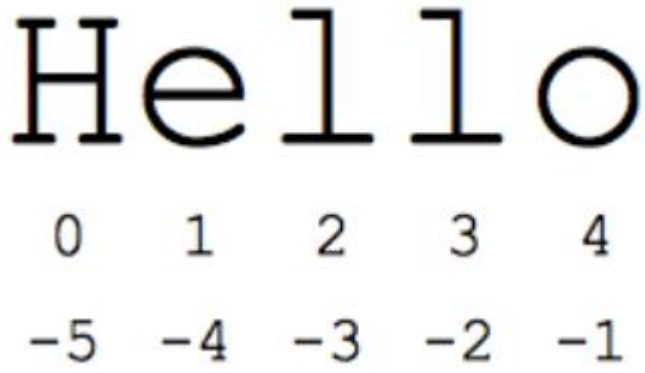
Example

```
name = 'Computer Programming'
first_word = name[0:8]
last_word = name[9:20]
print(first_word)
print(last_word)
```

Note: The last digit is not inclusive

```
Computer
Programming
```

Example



Example

- **s[-1] is 'o'** -- last char (1st from the end)
- **s[-4] is 'e'** -- 4th from the end
- **s[:-3] is 'He'** -- going up to but not including the last 3 chars.
- **s[-3:] is 'llo'** -- starting with the 3rd char from the end and extending to the end of the string.

String Methods

String Methods

- A method is similar to a function.
- Methods can be accessed by using a **dot .** Operator.

Lower-case and Upper-case

- Imagine if we want to convert a string to either upper case or lower case, we can use the methods called lower() or upper() e.g.,

```
s = "hello"
print(s.upper())
```

```
HELLO
```

Membership operator

Membership operators

- Membership operators are used for searching whether a specific character is part/member of a given string or not.

in operator

- “a” in the string:
- Returns boolean True if “a” is in the string
- Returns False if “a” is not in the string.

In operator

```
txt = "I love MLAD program"
print("MLAD" in txt)
```

```
True
```

In operator

```
txt = "The best things in life are free!"
if "free" in txt:
  print("Yes, 'free' is present.")
```

```
Yes, 'free' is present.
```

not in operator

- “a” not in the string:
- Returns boolean True if “a” is not in the string
- Returns False if “a” is in the string

Not operator

```
txt = "The best things in life are free!"
print("expensive" not in txt)
```

```
True
```

Not operator

```
txt = "The best things in life are free!"
if "expensive" not in txt:
  print("No, 'expensive' is NOT present.")
```

```
No, 'expensive' is NOT present.
```

Membership operators

These operators are also useful to find whether a specific substring is part of a given string

Isalnum()

Isalnum()

- The isalnum() method returns True if all characters in the string are alphanumeric (either alphabets or numbers)

- If not, it returns False

Syntax

string.isalnum()

the isalnum() method checks if all the characters of string are alphanumeric or not

Isalnum()

```
txt = "Company12"

x = txt.isalnum()

print(x)
```

```
True
```

upper()

upper()

- The upper() method returns the uppercased string from the given string
- It converts all lowercase characters to uppercase
- If no lowercase characters exist, it returns the original string

Syntax

string.upper()

Parameters:
upper() does not take any parameters

Example

school = ‘norquest’
school.upper()
Output: NORQUEST

school = ‘Norquest’
school.upper()
Output: NORQUEST

lower()

lower()

- The lower() method returns the lowercased string from the given string
- It converts all uppercase characters to lowercase
- If no uppercase characters exist, it returns the original string

Syntax

string.lower()

Parameters:
lower() does not take any parameters

Example

```
school = ‘NORQUEST’
school.lower()
Output: norquest

school = ‘Norquest’
school.lower()
Output: norquest
```

find()

find()

- The find() method returns the index of first occurrence of the substring (if found)

- If not found, it returns -1.

Syntax

s.find('other’)

- searches for the given other string within s
- returns the first index where it begins or -1 if not found

Syntax

```
txt = "Hello, welcome to my world."

x = txt.find("welcome")

print(x)
```

```
7
```

Questions?

References

1. https://greenteapress.com/wp/think-python-2e/
2. https://www.programiz.com/python-programming/methods/string/isalnum
3. https://www.geeksforgeeks.org/isupper-islower-lower-upper-python-applications/

NORQUEST COLLEGE